



# Desarrollo y gestión ágil de proyectos Python

**Lección 1: Desarrollo y gestión de  
proyectos**

# ÍNDICE

Desarrollo y gestión de proyectos.....	2
Presentación y objetivos .....	2
<b>1. Desarrollo y gestión de proyectos .....</b>	<b>3</b>
1.1. ¿Qué es un proyecto?.....	3
1.2. ¿Qué es la gestión de proyectos? .....	4
<b>2. Modelos de gestión y desarrollo de proyectos .....</b>	<b>6</b>
2.1 Modelo en Cascada .....	9
2.1.1 Métrica v3 .....	9
2.2. PMP .....	12
2.3. Agile.....	14
2.4. Scrum.....	15
2.5. Kanban.....	16
<b>3. Herramientas para la gestión de proyectos .....</b>	<b>18</b>
3.1. Asana .....	19
3.2. Trello .....	21
3.3. Redmine .....	22
3.4. Jira.....	24
<b>4. Puntos clave.....</b>	<b>27</b>

# Desarrollo y gestión de proyectos

*«Un objetivo sin un plan es sólo un deseo» ~ Larry Elder*

## PRESENTACIÓN Y OBJETIVOS

Desde los inicios del desarrollo de software se hizo evidente la necesidad de una buena gestión y planificación para conseguir los objetivos deseados, surgiendo la ingeniería del software.

La Ingeniería de Software es una de las ramas de la informática que estudia la creación de software fiable y de calidad, basándose en métodos y técnicas de ingeniería.

Todos los ingenieros de software pueden programar, pero no todos los programadores pueden gestionar un proyecto software.

Esta asignatura te ayudará a convertirte no solo en programador, sino que también te preparas para poder convertirte en ingenier@ de software y esta primera lección sentará las bases sobre las que trabajaremos en el resto de la asignatura.



### Objetivos

En esta lección aprenderás:

- Los conceptos básicos relacionados con los proyectos.
- Los modelos clásicos y actuales de desarrollo y gestión de proyectos.
- Herramientas para la gestión de proyectos.

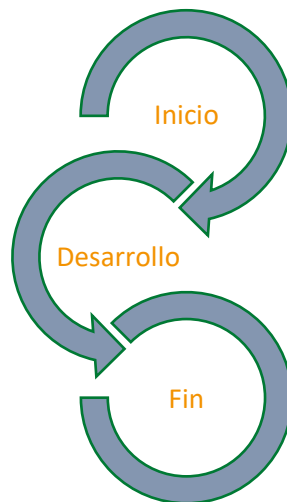
## 1. DESARROLLO Y GESTIÓN DE PROYECTOS

### 1.1. ¿Qué es un proyecto?

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

Tiene un principio y un final definidos.

Se considera finalizado cuando se han llevado a cabo los objetivos, cuando no es posible que se cumplan los objetivos o cuando no existe la necesidad que inició el proyecto.



*Ilustración 1.1. Representación de proyecto*

Todo proyecto crea un producto, servicio o resultado único.

Los proyectos pueden tener impactos sociales, económicos y ambientales que durarán mucho más que los propios proyectos.

Se llevan a cabo en todos los niveles de una organización.

Pueden involucrar a una sola persona, una sola unidad o múltiples unidades dentro de la organización.

Un proyecto puede generar:

- Un producto que puede ser un componente de otro elemento o un elemento final en sí mismo.
- La capacidad de realizar un servicio.
- Un resultado tal como un producto o un documento.

## 1.2. ¿Qué es la gestión de proyectos?

En todo proyecto hay un equipo de desarrollo con diversos roles y uno de esos roles es de gestión del propio proyecto. Este rol es desempeñado por una figura con una amplia experiencia y dotes de gestión y organizativos.

La dirección o gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo.

Se logra mediante la aplicación e integración adecuadas de una serie de procesos agrupados, que conforman cinco grupos de procesos.

Estos grupos de procesos son:

1. Inicio
2. Planificación
3. Ejecución
4. Seguimiento y Control
5. Cierre

Dirigir un proyecto implica:

- | Identificar requisitos
- | Abordar las necesidades, inquietudes y expectativas de los interesados según se planifica y efectúa el proyecto
- | Equilibrar las restricciones contrapuestas del proyecto que se relacionan.

Existen muchos aspectos y restricciones a la hora de gestionar un proyecto:

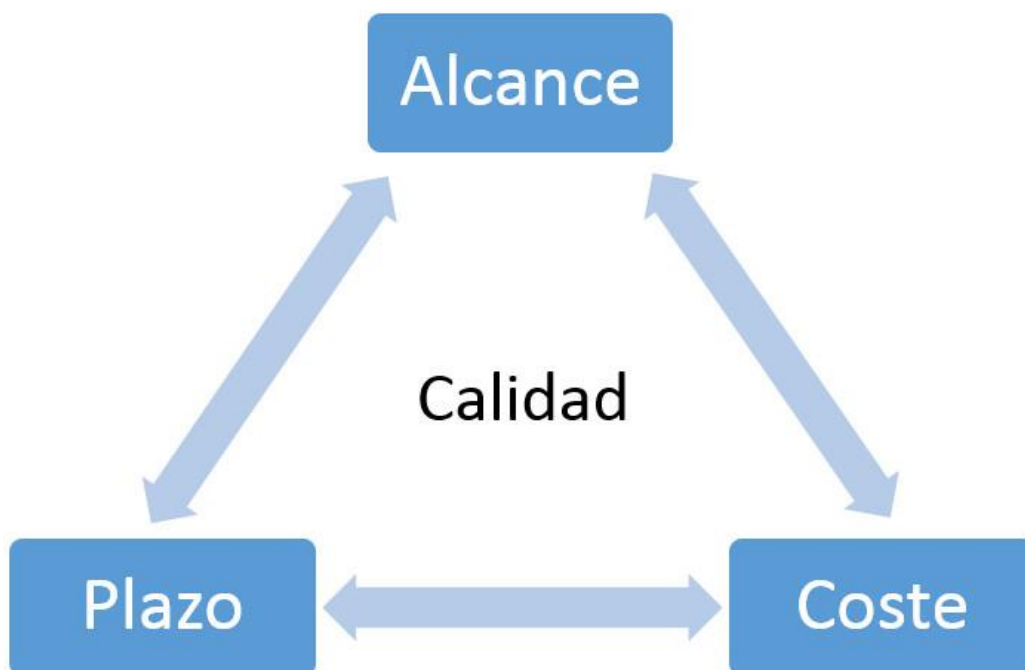
- El alcance
- La calidad
- El tiempo
- El presupuesto
- Los recursos
- El riesgo

Sin embargo, las tres más comunes son el tiempo, el coste y el alcance.

- | El alcance es importante para especificar todos los pasos del desarrollo del proyecto.
- | El tiempo consiste en mantener el proyecto dentro del cronograma y alcanzar los objetivos deseados.
- | El coste está compuesto por un presupuesto establecido en la etapa inicial del proyecto.

Estas tres restricciones están interconectadas y dependen mucho la una de la otra. Es lo que se conoce como el **Triángulo de Gestión de Proyectos**.

Una vez que se reduce el tiempo asignado para el proyecto, el costo aumenta. Además, el alcance del proyecto dicta el ritmo y una serie de recursos necesarios para realizar y completar con éxito el proyecto.



*Ilustración 1.2. Triángulo de Gestión de Proyectos*

## 2. MODELOS DE GESTIÓN Y DESARROLLO DE PROYECTOS

En 1968, durante la llamada “crisis del software”, la organización OTAN celebró la primera conferencia centrada en analizar los problemas de la programación.

Se puso de manifiesto la necesidad de crear una disciplina científica que permitiera aplicar un enfoque sistemático y cuantificable al desarrollo, operación y mantenimiento de sistemas informáticos. Esto se tradujo en el intento de aplicar ingeniería de procesos al software, surgiendo así la “ingeniería del software”.



### ***Presta atención***

Esta primera estrategia (tesis) se basó en dos pilares:

- Ingeniería de procesos: en los entornos de producción industrial existía un principio básico de calidad, contrastado con éxito: «la calidad del resultado depende de la calidad de los procesos empleados». Dicho de otra forma: no se necesita a gente brillante o muy cualificada; mientras los procesos empleados sean de calidad, el resultado será de calidad.
- Gestión predictiva: un tipo de gestión que se centra en garantizar el cumplimiento de agendas y presupuestos.

Desde los 80 se han desarrollado tantos modelos y prácticas para mejorar la calidad y eficiencia de los proyectos que puede resultar abrumador, ya que mientras la disciplina evolucionaba y se perfeccionaba a través de diferentes modelos de procesos y cuerpos de conocimiento para gestión de proyectos (MILQ9858, ISO9000, ISO9000-3, ISO 12207, SPICE, SW-CMM...) en la industria del software surgían dudas y se cuestionaba esta estrategia.

Desde mediados de los 90 hasta 2005-2010 han sido habituales las posturas radicales entre los defensores de los modelos de procesos (tesis) y de los marcos ágiles (antítesis).

No estaba claro que la planificación predictiva fuese apropiada para cualquier proyecto. En la práctica puede verse que a veces los criterios del éxito no son siempre el cumplimiento de fechas, costes y funcionalidades preestablecidas.

Por otra parte, también se cuestiona si en el desarrollo de software, como en otros trabajos basados en el conocimiento, se puede producir con patrones de procesos industriales. Es decir, se empieza a aceptar que el conocimiento tácito de la persona que realiza el trabajo puede aportar más al valor del resultado que la tecnología y los procesos empleados.

El **desarrollo de un proyecto** puede darse de forma completa o incremental.

- En el caso de un desarrollo completo, la descripción de lo que se desea obtener está disponible al inicio del proyecto; es completa y detallada y sirve de base para estimar. Con el plan inicial se organizan tareas, recursos y agenda de trabajo. Durante la ejecución se gestiona el cumplimiento de lo que se ha previsto.
- En los desarrollos incrementales, la descripción completa de lo que se desea obtener no está disponible al inicio. Se complementa y evoluciona durante el desarrollo, que se puede gestionar con dos tácticas diferentes:
  - Desarrollo incremental continuo: Empleando técnicas para lograr un flujo continuo de desarrollo de las funcionalidades o partes del producto, que se entregan de forma continua al cliente.
  - Desarrollo iterativo: Empleando técnicas de tiempo prefijado o "*timeboxing*" para mantener la producción de incrementos del producto a un ritmo fijo. Como veremos más adelante, éste es el marco de producción empleado al aplicar el marco estándar de scrum, que define como *Sprint* a cada iteración de desarrollo, al final de la cual se produce un incremento del producto: una parte entregable y lista para usarse.



La forma de trabajar puede ser **secuencial («en cascada»)** o **concurrente**.

El trabajo secuencial divide el trabajo en fases. Una fase nueva comienza cuando se termina la anterior. El ejemplo más habitual es el ciclo de cascada definido en ingeniería del software, cuyas fases son: definición de requisitos, análisis, diseño, codificación, pruebas e implementación.

Trabajar de forma concurrente significa solapar en el tiempo las diferentes fases. Siguiendo con el ejemplo de ingeniería de software, todas las fases del párrafo anterior se revisarían de forma simultánea y continua.

Una vez aclarados los conceptos de proyecto, gestión y desarrollo de proyectos y la historia detrás de los mismos, queda claro que existen diversos marcos de trabajo o modelos de gestión de proyectos, dentro de ellas podemos distinguir dos grandes grupos, los modelos clásicos (modelos de procesos o tesis) y los marcos de trabajo basado en agilidad.



### **Conceptos**

De entre todos los marcos de trabajo o modelos surgidos a lo largo de la historia de desarrollo de software vamos a destacar los que más trascendencia han tenido.

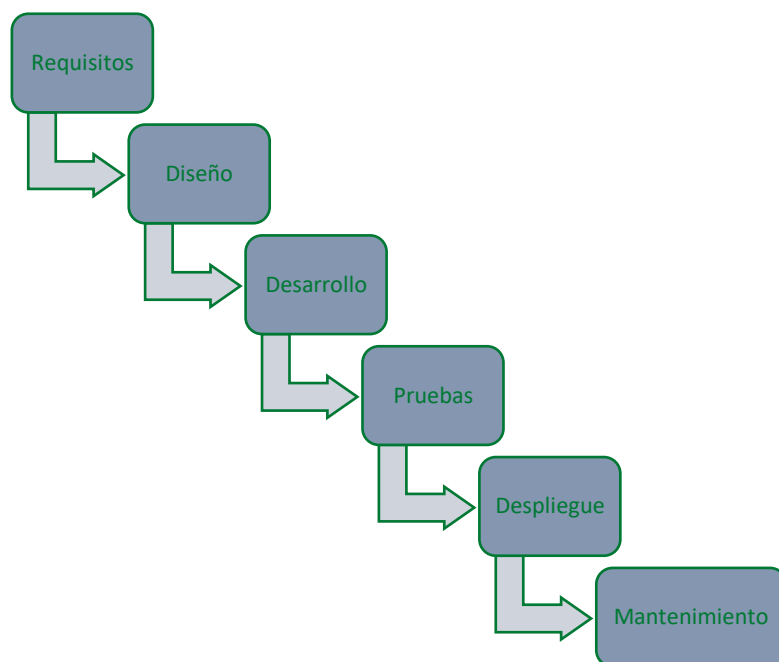
## 2.1 Modelo en Cascada

El modelo en cascada es el método tradicional de gestión de proyectos. Es un modelo secuencial, es decir, durante un proyecto en cascada, todo se hace en una secuencia planificada y predeterminada.

En primer lugar, se conceptualiza y planifica el proyecto, después de lo cual el desarrollo comienza con la implementación de la garantía de calidad.

Una vez que el proyecto ha llegado a su conclusión, se realiza el mantenimiento.

Los requisitos del proyecto se establecen en las primeras etapas de la planificación y rara vez se modifican a menos que sea absolutamente necesario.



*Ilustración 2.1. Modelo en cascada*

### 2.1.1 Métrica v3

MÉTRICA es una metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por Ministerio de Administraciones Públicas del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas.

Métrica tiene tres versiones, datando de la última actualización de 2001.

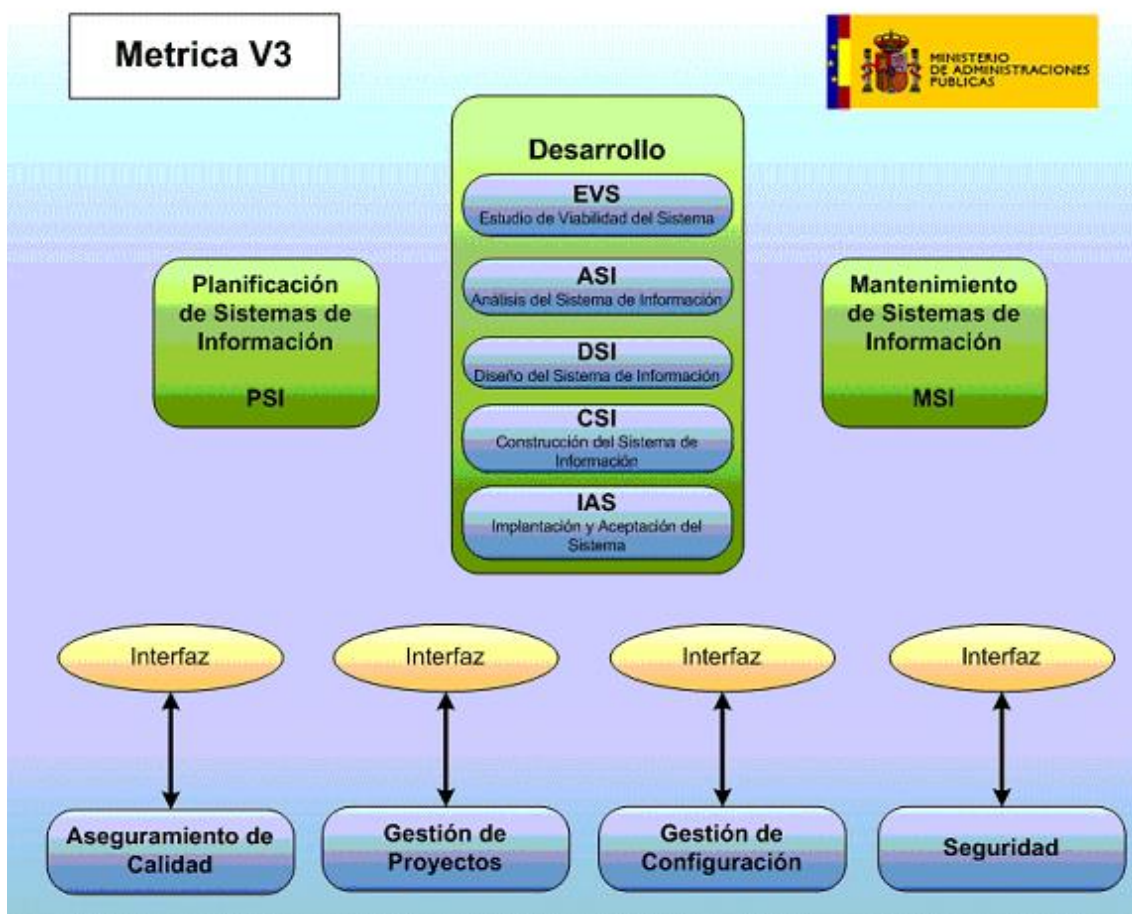
- Versión 1 - 1989
- Versión 2 - 1993
- Versión 2.1 - 1995
- Versión 3 - 2001

MÉTRICA está orientada al proceso y, en su versión 3, estos procesos son:

- Planificación de Sistemas de Información (PSI).
- Desarrollo de Sistemas de Información (DSI). Debido a su complejidad, está a su vez dividido en cinco procesos:
  - 1 Estudio de Viabilidad del Sistema (EVS).
  - 2 Análisis del Sistema de Información (ASI).
    - Se describe inicialmente el sistema de información.
    - Se recogen de forma detallada los requisitos funcionales.
    - Se identifican los subsistemas de análisis.
    - Se elaboran los modelos.
    - Se realiza un análisis de consistencia.
    - Se elabora el producto Especificación de Requisitos Software.
    - Se inicia la especificación del Plan de Pruebas.
  - 3 Diseño del Sistema de Información (DSI).
  - 4 Construcción del Sistema de Información (CSI).
  - 5 Implantación y Aceptación del Sistema (IAS).
    - Establecimiento del Plan de Implantación.
    - Formación Necesaria Para la Implantación.
    - Incorporación del Sistema al Entorno de Operación.
    - Carga de Datos al Entorno de Operación.
    - Pruebas de Implantación del Sistema.
    - Pruebas de Aceptación del Sistema.
    - Preparación del Mantenimiento del Sistema.
    - Establecimiento del Acuerdo de Nivel de Servicio.
    - Presentación y Aprobación del Sistema.
    - Paso a Producción.
- Mantenimiento de Sistemas de Información (MSI).

Métrica v3 era una de las metodologías de desarrollo software más ampliamente usadas hasta los años 2000.

Se encuadra dentro del modelo en cascada, ya que este enfoque metodológico ordena de forma rigurosa las etapas del proceso de desarrollo de modo que una no podía comenzar hasta concluir la anterior. Estas etapas quedaban así dispuestas de forma que el desarrollo iba “cayendo” de una fase a otra.



*Ilustración 2.2. Métrica v3*

Pese a ser un modelo poco evolucionado, hoy día, en algunas Administraciones Públicas se continúa trabajando con Métrica 3. Es una metodología definida por el Consejo Superior de Informática del Gobierno de España por lo que es habitual que se exija en numerosos contratos públicos.

Es una metodología muy rígida, que no permite cambios. En caso de que exista la necesidad de un cambio, éste suele ser muy costoso, cada vez más, a medida que avanza el desarrollo.

Si durante el desarrollo se detecta un error en el diseño o se entendió mal un requisito, habría que comenzar todo el trabajo desde el principio o rehacer gran parte de lo que se ha hecho.

De la misma manera, si al unir todas las piezas juntas en la fase de codificación, se percibe que durante las pruebas no se integran bien.

Es una fase muy tardía y ahora habría que comenzar todo el trabajo desde el principio.

## **2.2. PMP**

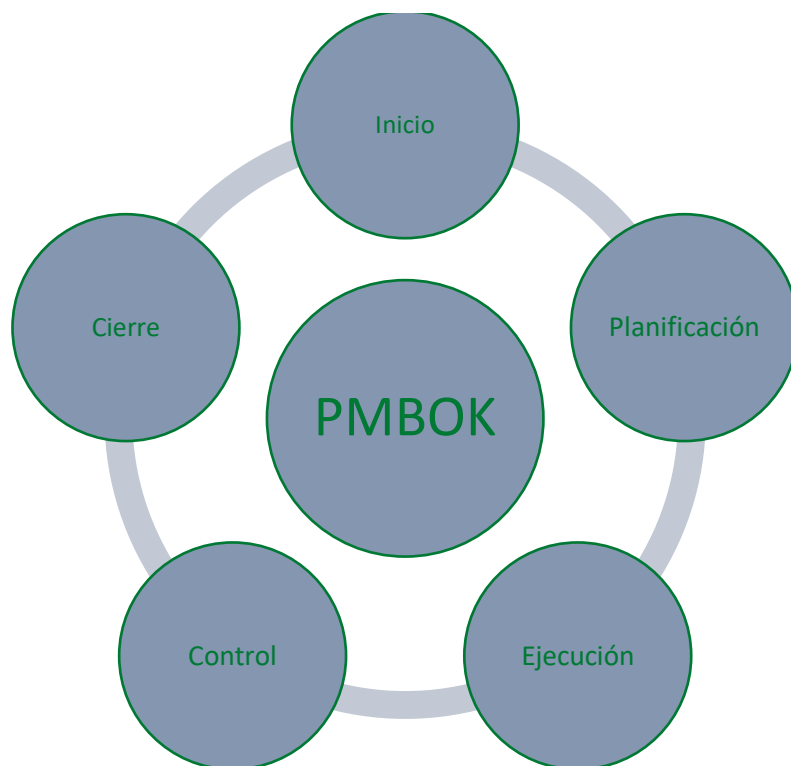
PMP (Project Management Professional) es un conjunto de estándares y convenciones establecidos en el PMBOK® (Project Management BodyOfKnowledge) para la gestión de proyectos, no es una metodología de gestión de proyectos propiamente dicha.

PMBOK contiene 5 grupos principales de procesos que se deben seguir durante un proyecto:

1. Grupo iniciador
2. Grupo de planificación
3. Grupo de ejecución
4. Grupo de control
5. Grupo de cierre

También sigue 10 áreas de conocimiento:

1. Gestión del tiempo
2. Gestión de costes
3. Gestión de comunicación
4. Gestión de recursos humanos
5. Gestión de la integración
6. Gestión de adquisiciones
7. Gestión de calidad
8. Gestión de riesgos
9. Gestión del alcance del proyecto
10. Gestión de partes interesadas del proyecto



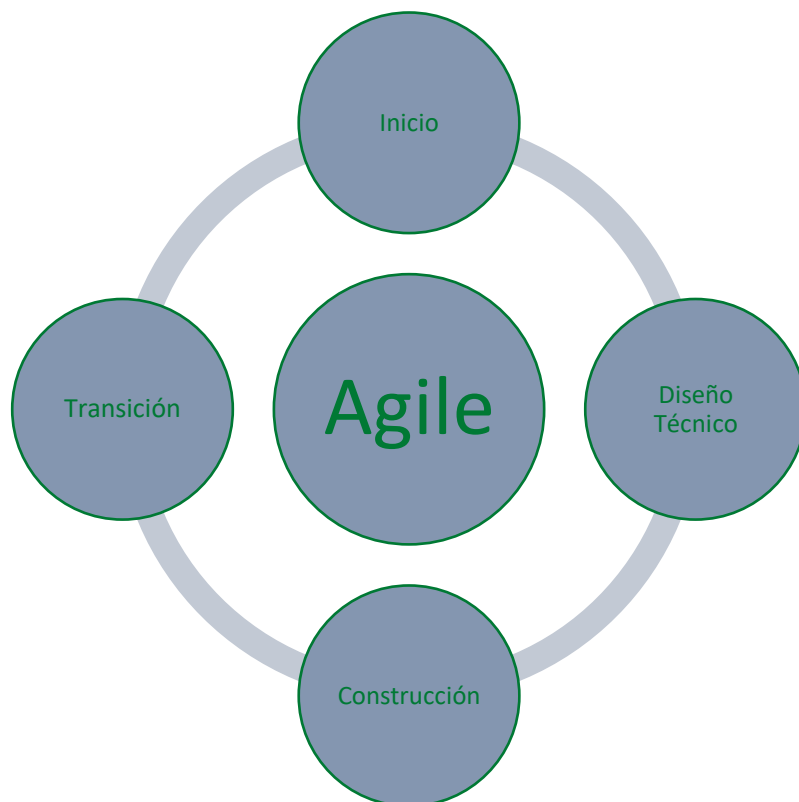
*Ilustración 2.3. PMP*

### 2.3. Agile

Durante un proyecto Agile hay un fuerte enfoque en la gestión del cambio: prepararse y adaptarse al cambio cuando, por ejemplo, hay un cambio en la asignación en el equipo del proyecto.

La retroalimentación también se considera un componente muy importante, ya sea comunicación entre los equipos que participan en el proyecto, retroalimentación del cliente o entre el jefe del proyecto y la gerencia.

Esto significa que todos tienen una comprensión firme del proceso de producción y la dirección en la que se dirige.



*Ilustración 2.4. Agile*

## 2.4. Scrum

Scrum es un framework para el desarrollo ágil, es decir, es un marco de trabajo que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Permite la creación de equipos auto organizados impulsando la colaboración de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

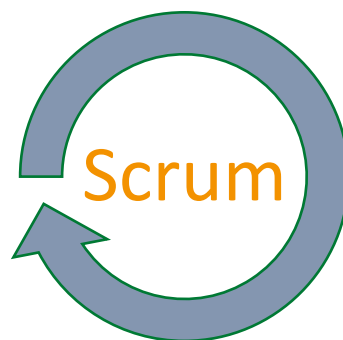
Se basa en el desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos.

Los requisitos se priorizan, es decir, se da prioridad a lo que tiene más valor para el cliente.

El equipo se sincroniza diariamente y se realizan las adaptaciones necesarias.

Tras cada iteración se muestra al cliente el resultado real obtenido, para que éste tome las decisiones necesarias en relación a lo observado.

Los equipos suelen ser pequeños (de 3 a 9 personas) y se le da la autoridad necesaria al equipo para poder cumplir los requisitos, siendo el propio equipo el que fija los tiempos máximos para lograr los objetivos.



*Ilustración 2.5. Scrum*

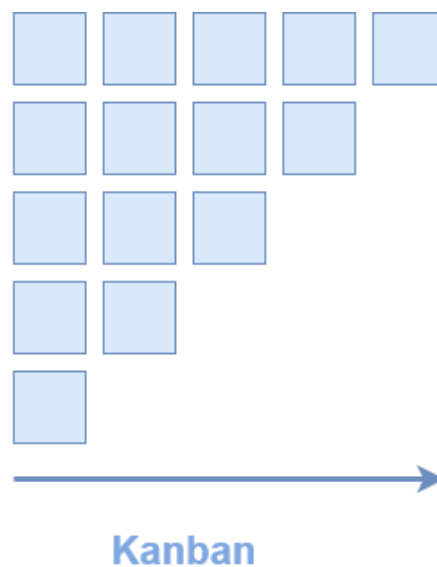


## 2.5. Kanban

Kanban es una metodología ágil que consiste en gestionar un proyecto de manera general, enfocándose en flujos de trabajo.

El método Kanban formulado por David J. Anderson es una aproximación al proceso gradual, evolutivo y al cambio de sistemas para las organizaciones.

Utiliza un sistema de extracción limitada del trabajo en curso como mecanismo básico para exponer los problemas de funcionamiento del sistema (o proceso) y estimular la colaboración para la mejora continua del sistema. Un ejemplo del sistema de extracción es el sistema Kanban, y es después de esta popular forma de trabajo en curso, que se ha denominado el método.



*Ilustración 2.6. Kanban*

El método Kanban tiene sus raíces en cuatro principios básicos:

1. Comience por lo que va a hacer ahora. No hay por qué empezar de 0 para aplicar el proceso Kanban. Puede comenzar aplicando el método Kanban en las funciones y procesos actuales y estimular cambios continuos, incrementales y evolutivos a su sistema.

2. Se acuerda perseguir el cambio incremental y evolutivo. La organización (o equipo) deben estar de acuerdo que el cambio continuo, gradual y evolutivo es la manera de hacer mejoras en el sistema y debe apegarse a ello. Los cambios radicales pueden parecer más eficaces, pero tienen una mayor tasa de fracaso debido a la resistencia y el miedo en la organización. El método Kanban anima a los pequeños y continuos cambios incrementales y evolutivos a su sistema actual.
3. Respetar el proceso actual, los roles, las responsabilidades y los cargos. Tenemos que facilitar el cambio futuro; acordando respetar los roles actuales, responsabilidades y cargos, eliminamos los temores iniciales. Esto nos debería permitir obtener un mayor apoyo a nuestra iniciativa Kanban.
4. Liderazgo en todos los niveles. En Kanban, el liderazgo no está relegado a unos pocos elegidos, todo lo contrario. Se debe alentar hechos de liderazgo en todos los niveles de la organización de los contribuyentes individuales a la alta dirección.

### 3. HERRAMIENTAS PARA LA GESTIÓN DE PROYECTOS

En esta lecciones presentamos algunas de las herramientas de gestión de proyectos más usadas, útiles y conocidas, pero hay muchas más tanto gratuitas como de pago.

¿Cuáles son las ventajas de las herramientas de gestión de tareas y proyectos online?

La respuesta es sencilla y tiene mucho que ver con la propia esencia de las nuevas tecnologías.

1. Permiten la gestión de tareas desde cualquier equipo o dispositivo móvil.
2. Impulsan una mejor comunicación entre los miembros de un equipo sobre aspectos concretos de una tarea.
3. Gestionan con agilidad y a través de recordatorios aspectos como los plazos de entrega o nuevas tareas.
4. Permiten incorporar documentos y compartirlos online con todos los miembros del equipo que participan en el proyecto.
5. Favorecen el debate entre las personas que comparten una tarea y la aportación de nuevas ideas que quedan reflejadas por escrito y no se pierden en reuniones.
6. Se centran solo en la tarea que se tiene que desarrollar y evitan las distracciones con otros temas. Mejora de la productividad.
7. Permiten evaluar la gestión de las tareas y proyectos y el grado de participación de todos los usuarios de un equipo gracias a la opción de realizar informes que muchas de ellas incorporan.
8. Permiten conocer en tiempo real cuál es el grado de ejecución de una tarea y tomar decisiones en caso de que ésta no se ajuste a los plazos de entrega o a los objetivos marcados.
9. Son especialmente útiles tanto para la gestión interna, como para la coordinación de acciones web. De la misma manera, este tipo de herramientas son la mejor opción para el seguimiento y desarrollo de una campaña de marketing digital.
10. Existen múltiples opciones gratis y en español de software para la gestión de proyectos online.

### 3.1. Asana

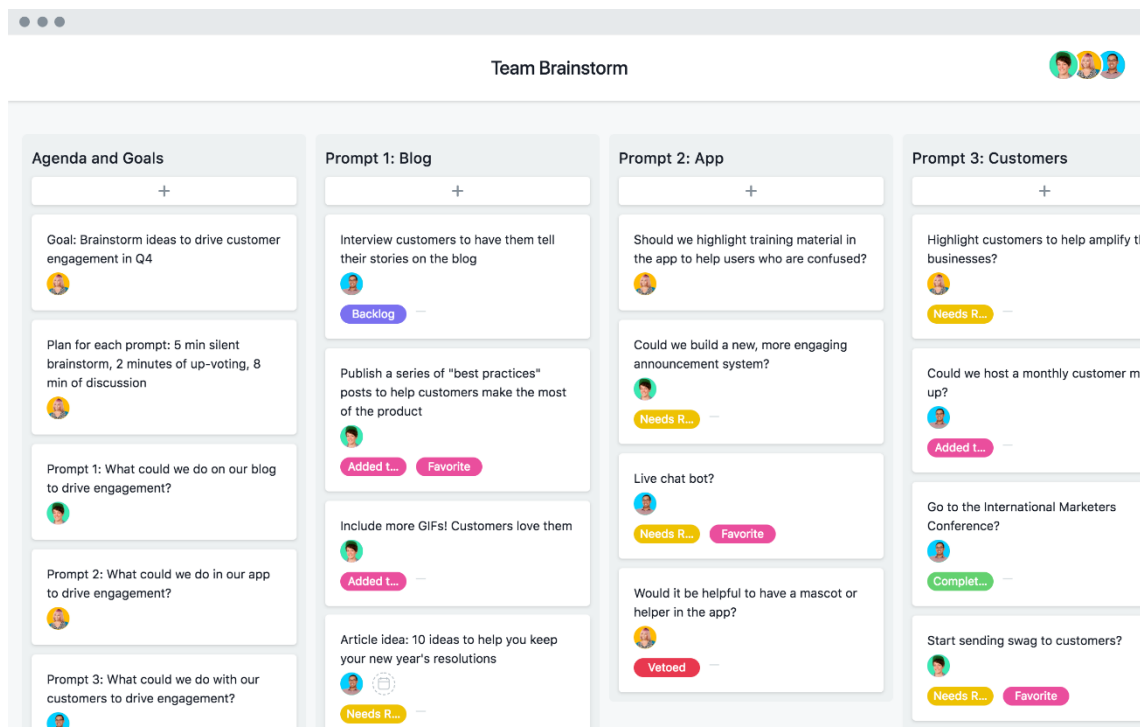
Asana es una aplicación web y móvil diseñada para mejorar la comunicación y colaboración en equipo. Fue desarrollada por el co-fundador de Facebook Dustin Moskovitz y el ingeniero Justin Rosenstein, que tenían, entre otras ocupaciones, mejorar la productividad de los empleados de Facebook.



*Ilustración 3.1. Logo Asana*

Asana es una aplicación colaborativa para la gestión de tareas. El producto tiene muchas funcionalidades, como espacios de trabajo, proyectos, proyectos personales, tareas, etiquetas, notas, comentarios y un buzón que organiza y actualiza la información en tiempo real.

El producto está diseñado para facilitar a las personas y a los equipos la planificación y la gestión de sus proyectos y tareas. Cada equipo tiene un espacio de trabajo. Los espacios de trabajo contienen proyectos y los proyectos contienen tareas.



*Ilustración 3.2.. Ejemplo Asana*

Moskovitz y Rosenstein piensan que “Las tareas son unidades atómicas de trabajo.”

En cada tarea, los usuarios pueden añadir notas, comentarios, ficheros y etiquetas.


Los usuarios pueden seguir proyectos y tareas, y cuando cambia el estado de un proyecto o una tarea, los seguidores tienen la información de dichos cambios en sus buzones.

Moskovitz y Rosenstein piensan que las empresas necesitan “una sola versión de la verdad” sobre lo que los empleados están haciendo.

En noviembre de 2011, según un artículo de Bloomberg Businessweek, el producto se describía como “un único sitio para saber en qué están trabajando los miembros del equipo de un proyecto, responder dudas, y obtener actualizaciones instantáneas sobre el progreso del trabajo.”

Asana es gratis para equipos de hasta 15 miembros. En abril de 2012, Asana lanzó Premium Workspaces, la versión de pago del producto, para equipos con más de 15 miembros.

En mayo de 2013, Asana lanzó Organizations, que permite que las empresas adopten Asana a nivel corporativo: desde cientos a miles de empleados. Con el lanzamiento de Organizations vinieron nuevas funcionalidades, entre otras: Asana Team Browser, una vista unificada de las tareas y los buzones de las personas; Employee auto-join y capacidades de administración relacionadas con la gestión de permisos.



***Para más información***

Puedes visitar la web de la herramienta:

<https://asana.com>

### 3.2. Trello

Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y Android para organizar proyectos.

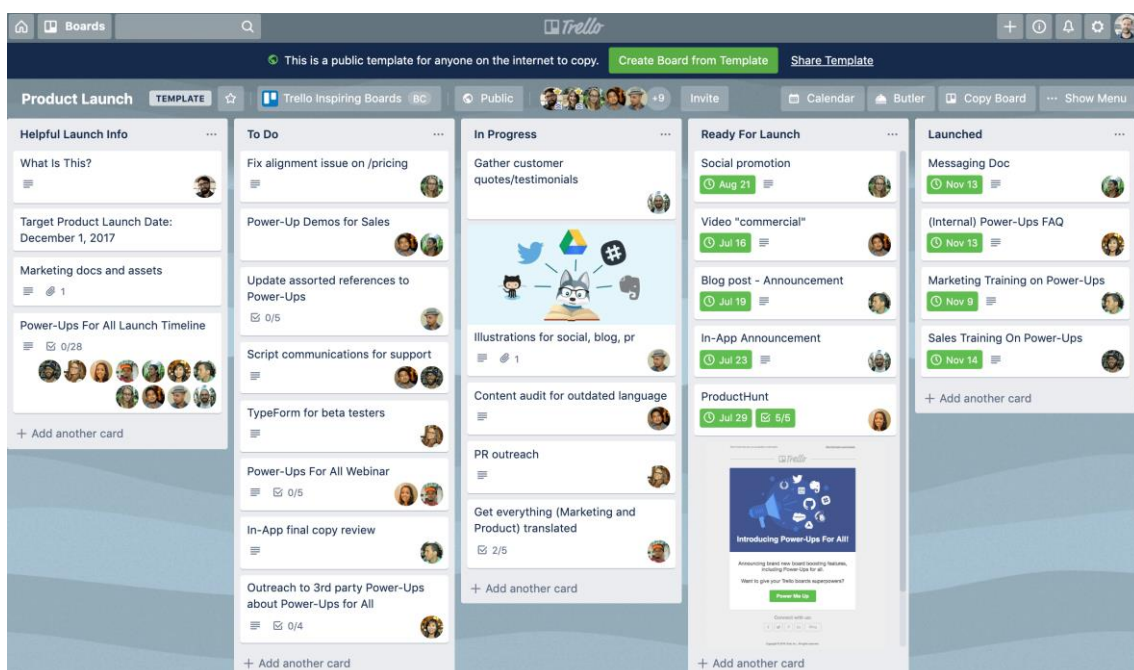


*Ilustración 3.3. Logo Trello*

Empleando el sistema kanban, para el registro de actividades con tarjetas virtuales organiza tareas, permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir tableros.

Trello ofrece un tablón virtual en el que se pueden colgar ideas, tareas, imágenes o enlaces.

Es versátil y fácil de usar pudiendo usarse para cualquier tipo de tarea que requiera organizar información.



*Ilustración 3.4. Ejemplo Trello*

Disponible en 21 idiomas, con interfaz web, clientes para iOS y Android.



**Para más información**

Puedes visitar la web de la herramienta:

<https://trello.com/es>

### 3.3. Redmine

Redmine es una herramienta para la gestión de proyectos, que con sus diversas funcionalidades permite a los usuarios de diferentes proyectos realizar el seguimiento y organización de los mismos.

Además, es posible optimizar su funcionamiento agregando funcionalidades. Incluye un sistema de seguimiento de incidentes con seguimiento de errores.

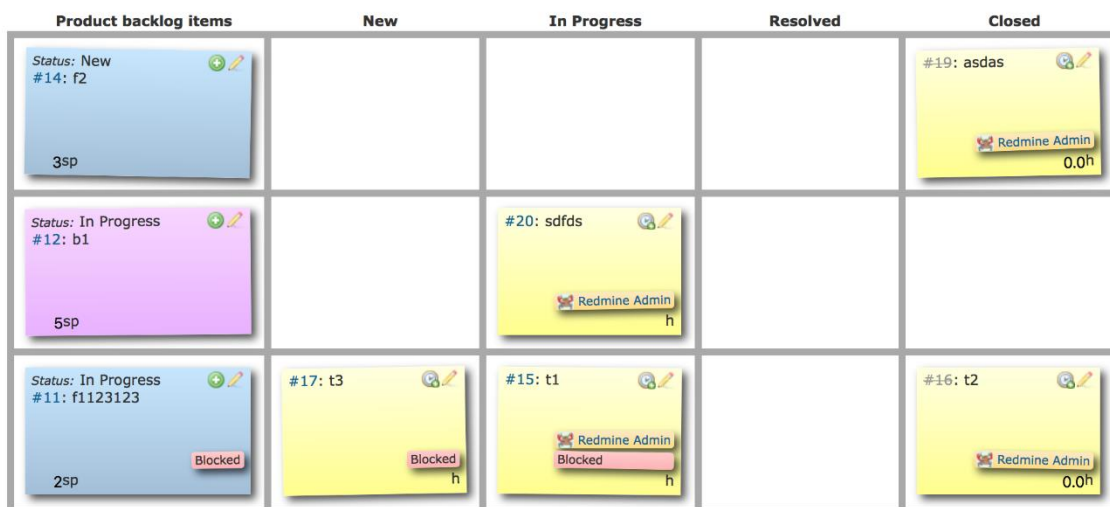
Otras herramientas que incluye son calendario de actividades, diagramas de Gantt para la representación visual de la línea del tiempo de los proyectos, wiki, foro, visor del repositorio de control de versiones, RSS, control de flujo de trabajo basado en roles, integración con correo electrónico, entre otras opciones.



*Ilustración 3.5. Logo Redmine*

Las características principales de Redmine son las siguientes:

- Soporta múltiples proyectos.
- Soporta múltiples usuarios.
- Roles flexibles basados en control de acceso.
- Sistema de seguimiento de errores flexible.
- Diagramas de Gantt y calendario.
- Administración de noticias, documentos y archivos.
- Fuentes web y notificaciones por correo electrónico.
- Creación de peticiones por correo.
- Soporte para autenticación con LDAP.
- Autoregistro de usuarios.
- Integración SCM (Subversion, CVS, Git, Mercurial, Bazaar y Darcs).
- Soporta múltiples plataformas (GNU/Linux, Microsoft Linux, Mac OSX)
- Soporta diferentes bases de datos (MySQL, PostgreSQL y SQLite).
- Extensiones: entre las que destaca el plugin para proyectos ágiles.



*Ilustración 3.6. Ejemplo Redmine*





**Para más información**

Puedes visitar la web de la herramienta:

<https://www.redmine.org/>

### 3.4. Jira

Jira es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos.

La herramienta fue desarrollada por la empresa australiana Atlassian.

Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estado de desarrollo y más tarde para la gestión de errores. Jira puede ser utilizado para la gestión y mejora de los procesos, gracias a sus funciones para la organización de flujos de trabajo.



*Ilustración 3.7.. Logo Jira*

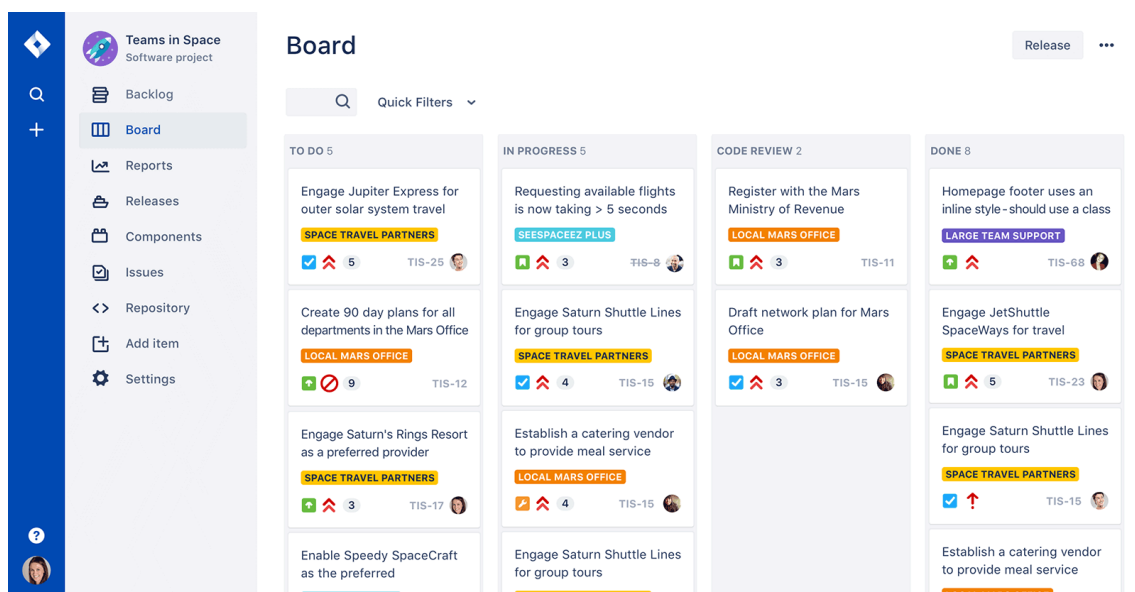
Jira Software es una herramienta de gestión de proyectos para equipos ágiles.

El marco de scrum permite a los equipos de software gestionar proyectos complejos creando una cultura de colaboración.

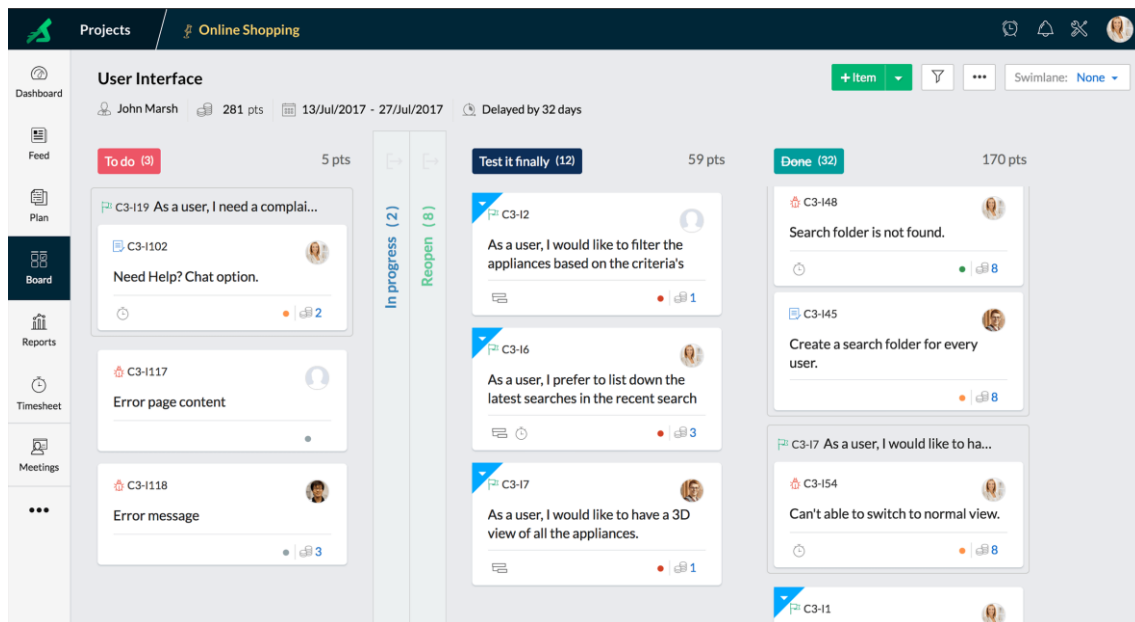
El tablero de Jira Scrum es la herramienta que une a los equipos en torno a objetivos únicos y promueve una entrega iterativa e incremental.

Si scrum es el marco que guía a los equipos y sus funciones, eventos, artefactos y reglas, el tablero de Jira Scrum es la visualización de su progreso durante el ciclo de desarrollo. El tablero de Jira Scrum es mucho más que un panel de tareas y entre sus funciones destacan:

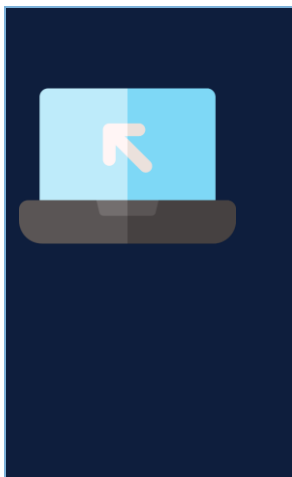
- Aumentar la comunicación y la transparencia. El tablero de Jira Scrum es la única fuente de verdad para el trabajo que el equipo debe realizar. Como a él pueden acceder todos los miembros del equipo en cualquier momento, todos saben exactamente lo que les corresponde hacer y pueden identificar rápidamente cualquier bloqueo.
- Promover la planificación de sprints y el desarrollo iterativo. En el núcleo del marco de scrum está el sprint, una cantidad designada de tiempo (generalmente dos semanas) para que los equipos creen un incremento de producto potencialmente publicable. El tablero de Jira Scrum se ha diseñado para que los equipos puedan organizar su trabajo en torno al plazo del sprint.
- Mejorar el enfoque y la organización del equipo. Los equipos no cumplen con los plazos del proyecto cuando están comprometidos en exceso con su carga de trabajo o pierden la pista de los hitos que son clave. Los tableros de Jira Scrum proporcionan transparencia en el trabajo del equipo al dividir el trabajo en etapas y al utilizar informes de trabajo pendiente y velocidad.



*Ilustración 3.8. Ejemplo tablero jira*



*Ilustración 3.9. Ejemplo tablero Jira en proyecto de desarrollo*



### **Para más información**

Puedes visitar la web de la herramienta:

<https://www.atlassian.com/es/software/jira>

## 4. PUNTOS CLAVE

- | Definición de proyecto y gestión de proyectos.
- | Modelos clásicos y ágiles de desarrollo y gestión de proyectos.
- | Herramientas para la gestión de proyectos

